EECE 5639 Computer Vision I

Lecture 16

Stereo: Epipolar Geometry, 3d Reconstruction

Motion

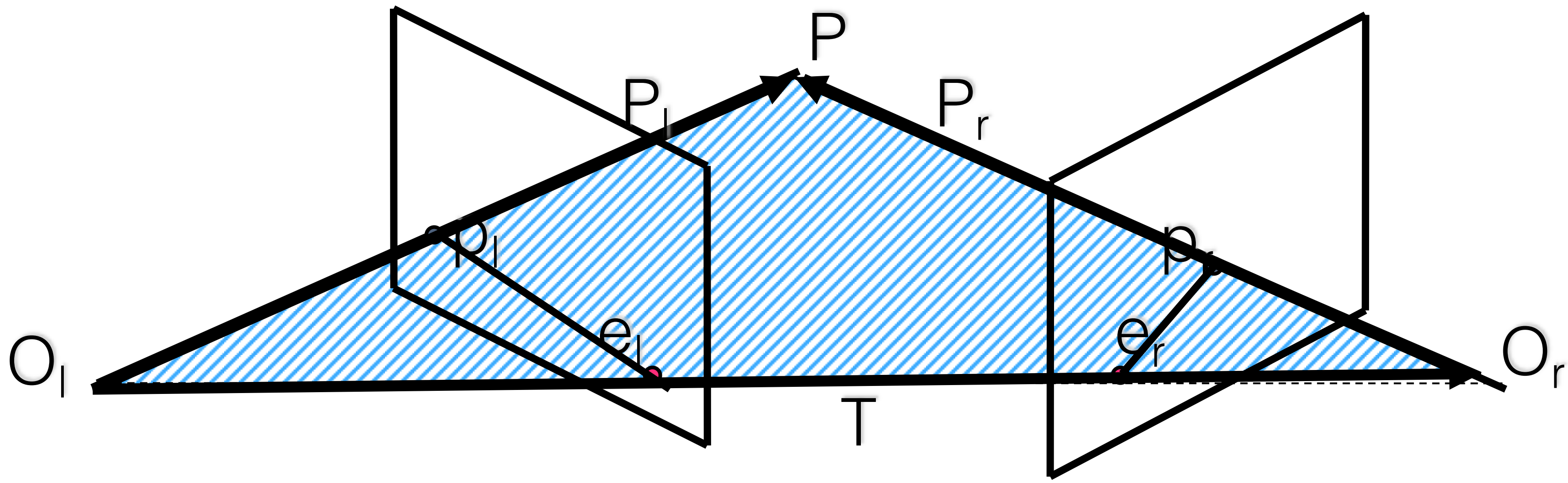
Hw 4 is out. Due March 29.

Next Class

- **Review Session**
- **2nd midterm NEW DATE: FRIDAY 3/25**
- **Topics:**
  - **Class 6 to 14 (up to, but not including stereo)**
  - **Hw 2 & 3**
  - **Project 2**

Essential Matrix

Epipolar constraint: P_l, T and P_l - T are coplanar:



$$P_r^T R S P_l = 0$$

Essential Matrix:

$$E = RS \qquad P_r^T E P_l = 0$$

Essential Matrix Summary

Longuet-Higgins equation $$p_r^T E p_l = 0$$

Epipolar lines:

$$\tilde{p_r}^T \tilde{l_r} = 0 \qquad \tilde{p_l}^T \tilde{l_l} = 0$$

$$\tilde{l_r} = E p_l \qquad \tilde{l_l} = E^T p_r$$

Epipoles:

$$e_r^T E = 0 \qquad E e_l = 0$$

Fundamental Matrix

The essential matrix uses **CAMERA** coordinates

To use **image coordinates** we must consider the INTRINSIC camera parameters:

$$\bar{p}_l = M_l p_l \qquad p_l = M_l^{-1} \bar{p}_l$$

$$\bar{p}_r = M_r p_r \qquad p_r = M_r^{-1} \bar{p}_r$$

Fundamental Matrix

$$p_l = M_l^{-1}\bar{p}_l$$

$$p_r = M_r^{-1}\bar{p}_r$$

$$p_r^T E p_l = 0$$

$$(M_r^{-1}\bar{p}_r)^T E (M_l^{-1}\bar{p}_l) = 0$$

$$\bar{p}_r^T (M_r^{-T} E M_l^{-1}) \bar{p}_l = 0$$

$$\boxed{\bar{p}_r^T F \bar{p}_l = 0}$$

Fundamental Matrix Properties

has rank 2

depends on the INTRINSIC and EXTRINSIC Parameters (f, etc ; R & T)

$$\bar{p}_r^{\,T} F \bar{p}_l = 0$$

$$F = M_r^{-T} R S M_l^{-1}$$

Analogous to the Essential matrix, the Fundamental matrix also tells how points in each image are related to epipolar lines in the other image.

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$\begin{gathered} x_i x'_i f_{11} + x_i y'_i f_{21} + x_i f_{31} + \\ y_i x'_i f_{12} + y_i y'_i f_{22} + y_i f_{32} + \\ x'_i f_{13} + y'_i f_{23} + f_{33} = 0 \end{gathered}$$

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

m correspondences:

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m x'_m & x_m y'_m & x_m & y_m x'_m & y_m y'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

$$A\mathbf{x} = 0$$

A has rank 8

$$\min_{\mathbf{x}} ||A\mathbf{x}||^2 \text{ s.t. } ||x||^2 = 1$$

- Find the eigenvector of A^TA with smallest eigenvalue!

Algorithm EIGHT_POINT

The input is formed by m point correspondences, m >= 8

Construct the m x 9 matrix A
Find the SVD of A: $A = UDV^T$
The columns of V are the eigenvectors of A^TA; the last one corresponds to the smallest eigenvalue:

The entries of F are the components of the last column of V corresponding to the least s.v.

Algorithm EIGHT_POINT

F must be singular. To enforce it:

Find the SVD of F: $F = U_f D_f V_f^T$

Set smallest s.v. of F to 0 to create D'_f

Recompute F: $F = U_f D'_f V_f^T$

Don’t Forget the Numerical Details:

The coordinates of corresponding points can have a wide range leading to numerical instabilities.

It is better to first normalize them so they have average 0 and stdev 1 and de-normalize F at the end:

$$\hat{x}_i = T x_i \qquad \hat{x}'_i = T' x'_i$$

$$T = \begin{bmatrix} \frac{1}{\sigma^2} & 0 & -\mu_x \\ 0 & \frac{1}{\sigma^2} & -\mu_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{F = T'^{T} F_n T}$$

Example

Example

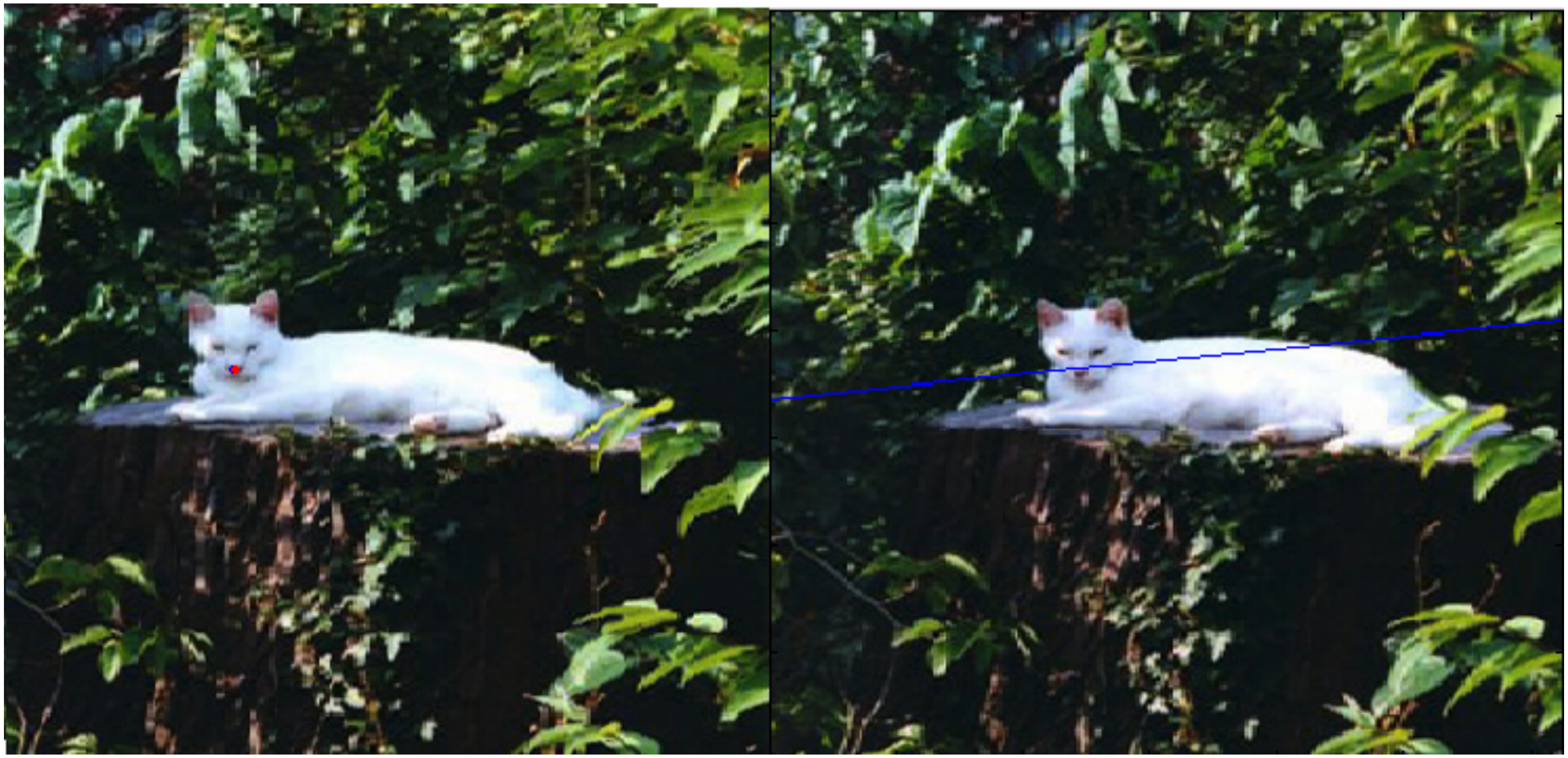


Image Rectification

Image Rectification

Assuming extrinsic parameters R & T are known, compute the image transformation that makes conjugate epipolar lines collinear and parallel to the horizontal image axis

Image Rectification

- Rectification involves two rotations:
  - First rotation sends epipole to infinity along horizontal axis
  - Second rotation makes epipolar lines parallel
- Rotate the left and right cameras with first R_1
- Rotate the right camera with the R matrix
- Adjust scales in both camera references

Image Rectification

Build the first rotation:

$$R_{rect} = \begin{bmatrix} \mathbf{e_1}^T \\ \mathbf{e_2}^T \\ \mathbf{e_3}^T \end{bmatrix}$$

with: $$\mathbf{e_1} = \frac{\mathbf{T}}{||\mathbf{T}||}$$ makes epipole go to the hor. axis

$$\mathbf{e_2} = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y \\ T_x \\ 0 \end{bmatrix}$$ orthogonal to optical axis

$$\mathbf{e_3} = \mathbf{e_1} \times \mathbf{e_2}$$

Algorithm Rectification

Build the matrix R_{rect}

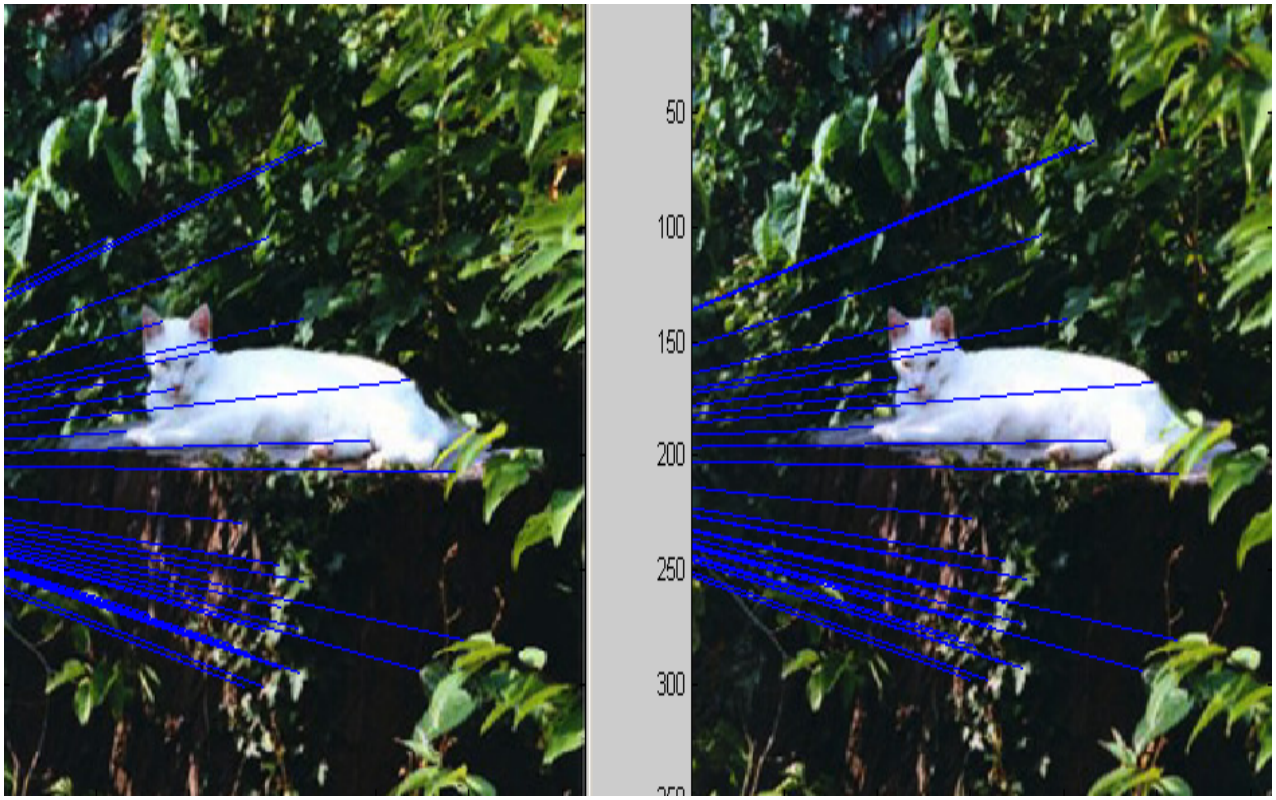
Set $R_l = R_{rect}$ and $R_r = R.R_{rect}$

For each left point $p_l = (x,y,f)^T$

- compute $R_l\, p_l = (x', y', z')^T$
- Compute $p'_l = f/z'\,(x', y'\, z')^T$

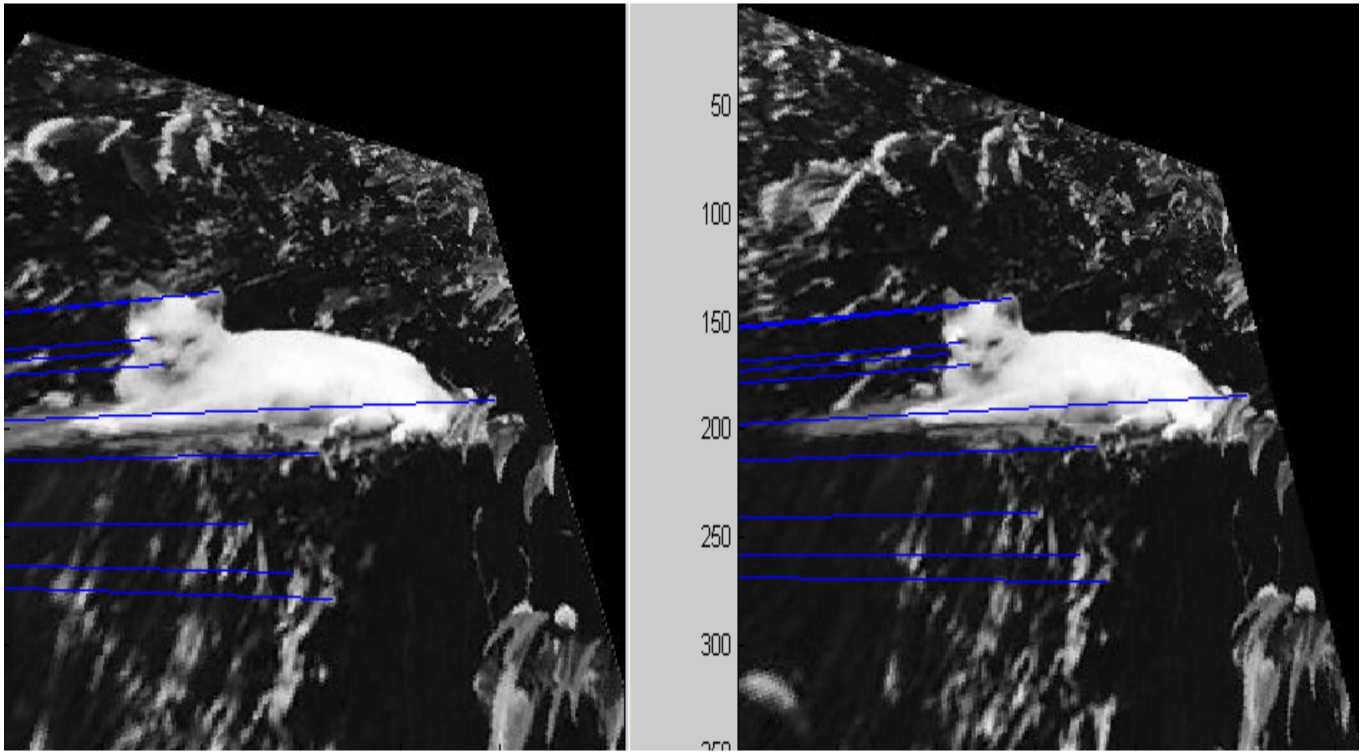
Repeat above for the right camera with R_r and p_r

Image Rectification



Stereo Images prior to rectification

Image rectification



Stereo Images after rectification

3D Reconstruction

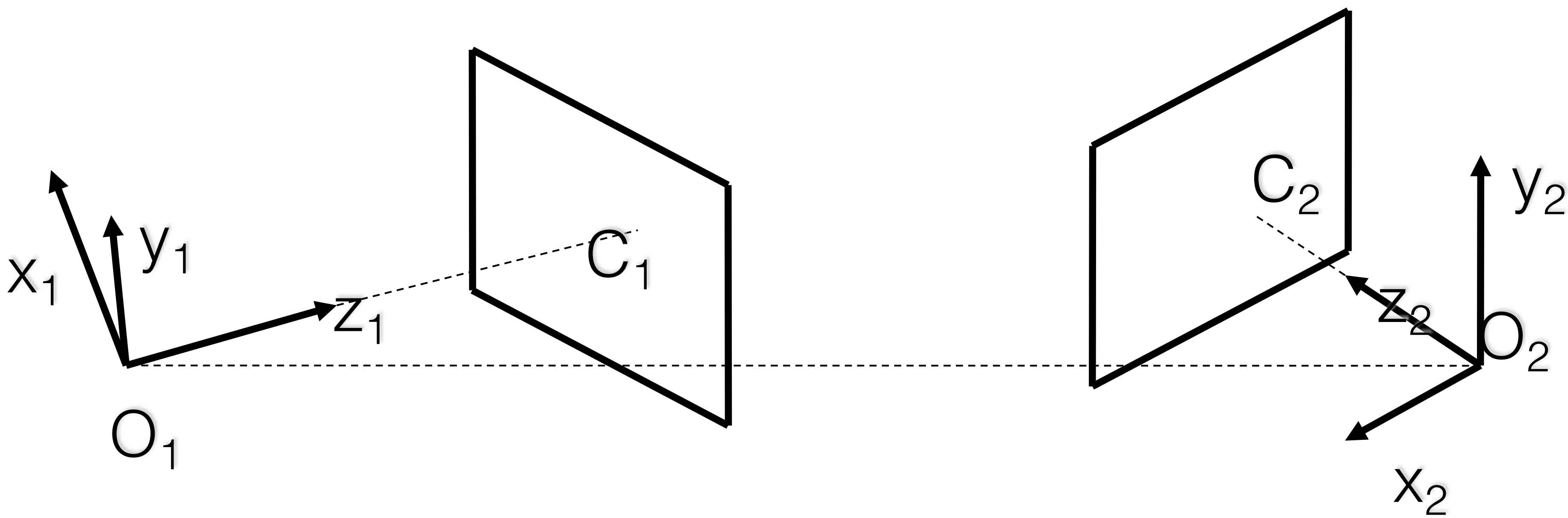
3D Reconstruction

3 cases:

- Fully calibrated: intrinsic & extrinsic parameters are known.
  - UNAMBIGUOUS reconstruction
- Only intrinsic parameters are known.
  - Up to a SCALE FACTOR
- No parameters are known.
  - Up to an UNKNOWN PROJECTIVE transformation

FULLY CALIBRATED

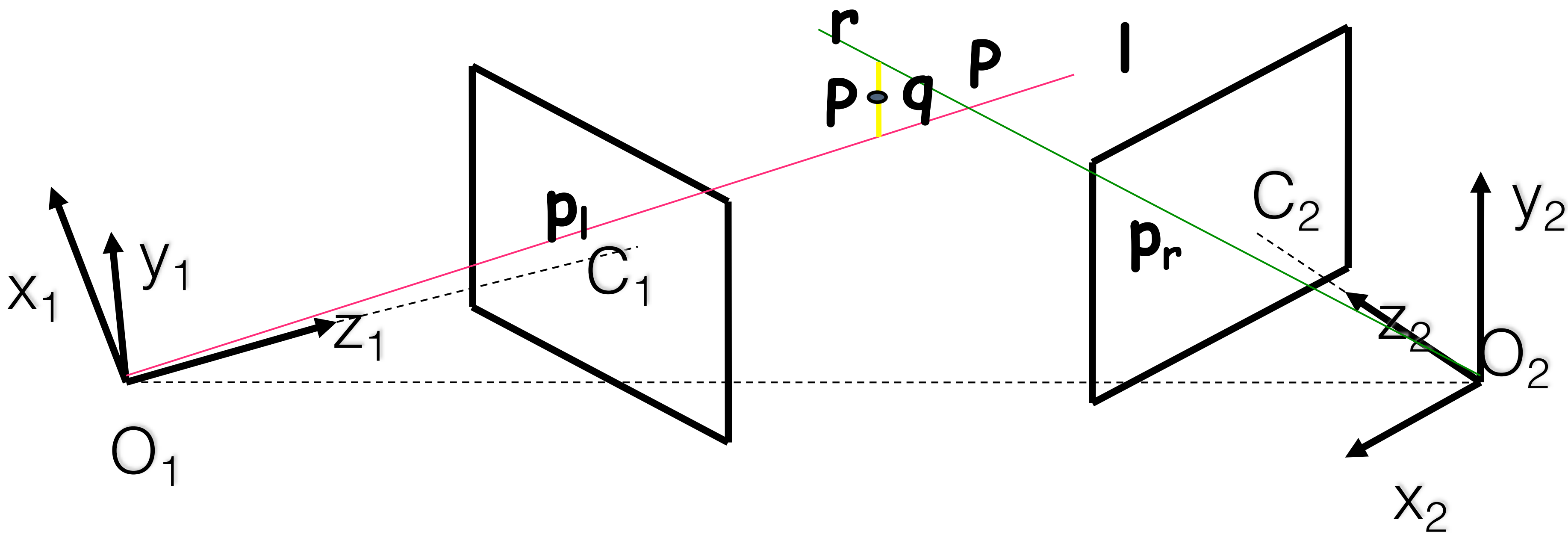
Parameters of a Stereo System



- Intrinsic:
- f_1 and f_2: focal lengths
- c_1 and c_2: principal points
- Pixel size

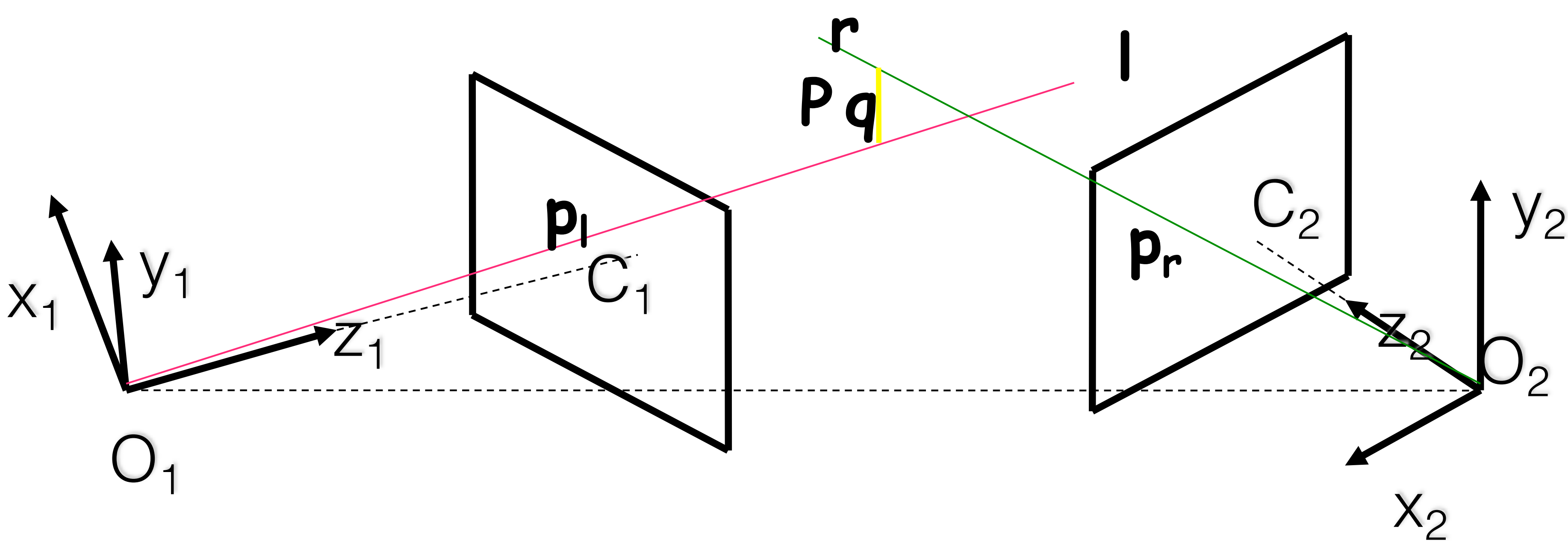
- Extrinsic
- Transformation (R,T) between cameras

3D Reconstruction



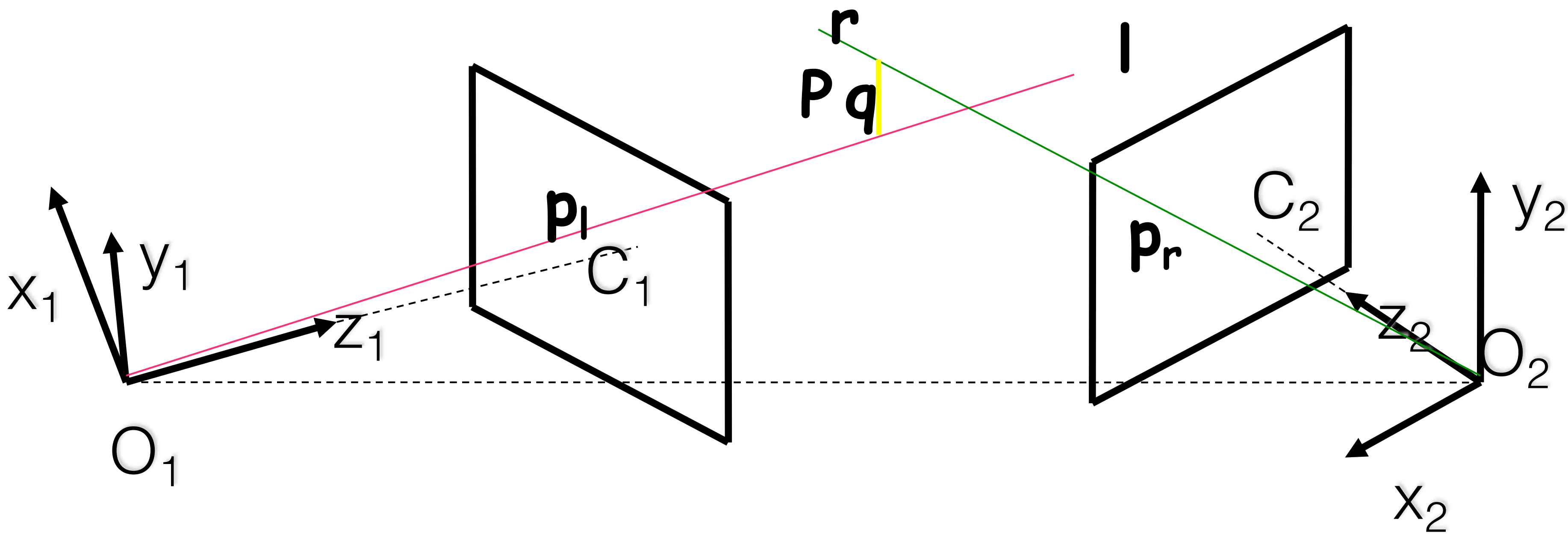
- P should lie in the intersection of the 2 rays r and l
- Due to numerical errors, correspondences errors, noise, etc, the rays may not intersect!

3D Reconstruction



$$
\begin{aligned}
T &= O_r - O_l \\
P_r &= R(P_l - T)
\end{aligned}
$$

3D Reconstruction



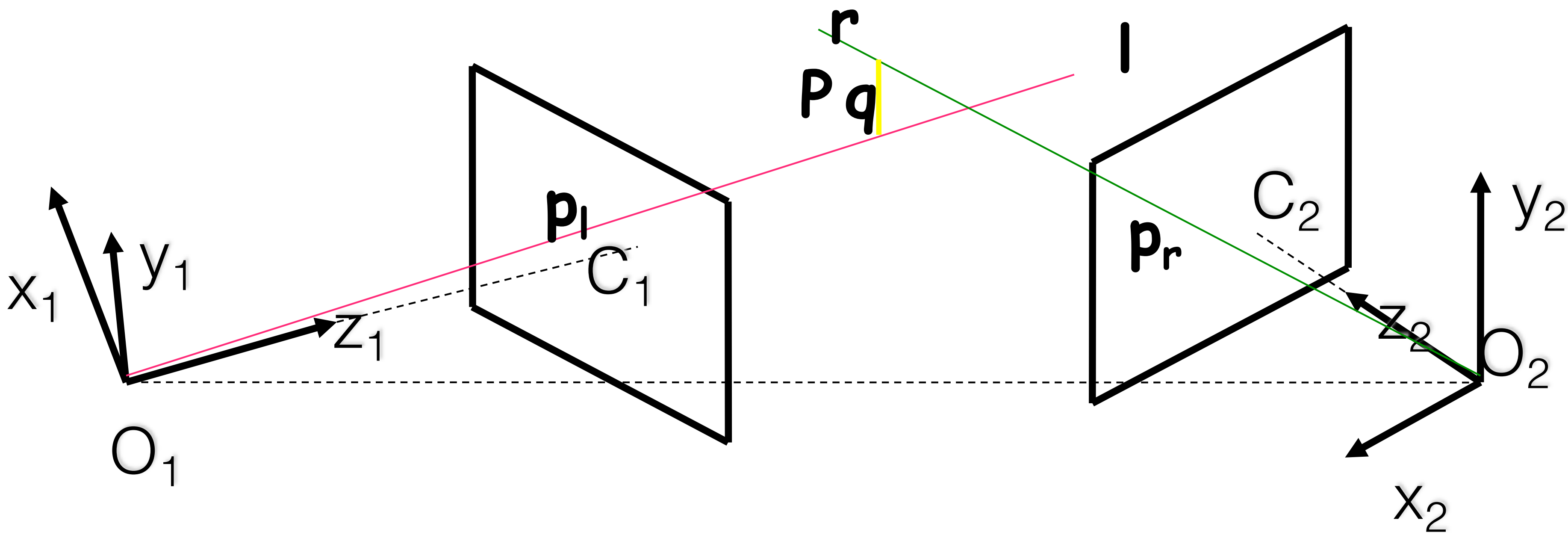
$$T = O_r - O_l$$
$$P_r = R(P_l - T)$$

Ray l:

$$p = a.p_l$$

a is a real number

3D Reconstruction



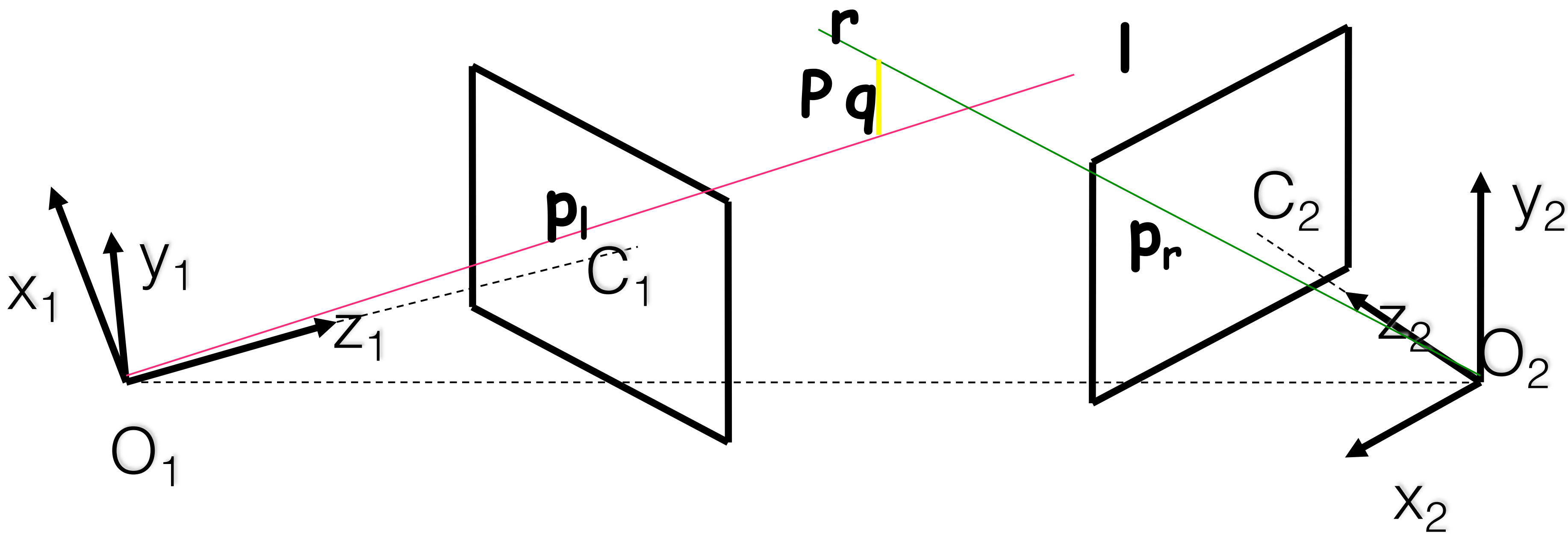
$$T = O_r - O_l$$
$$P_r = R(P_l - T)$$

Ray r:

$$p = T + b.R^T p_r$$

b is a real number

3D Reconstruction



Ray l:

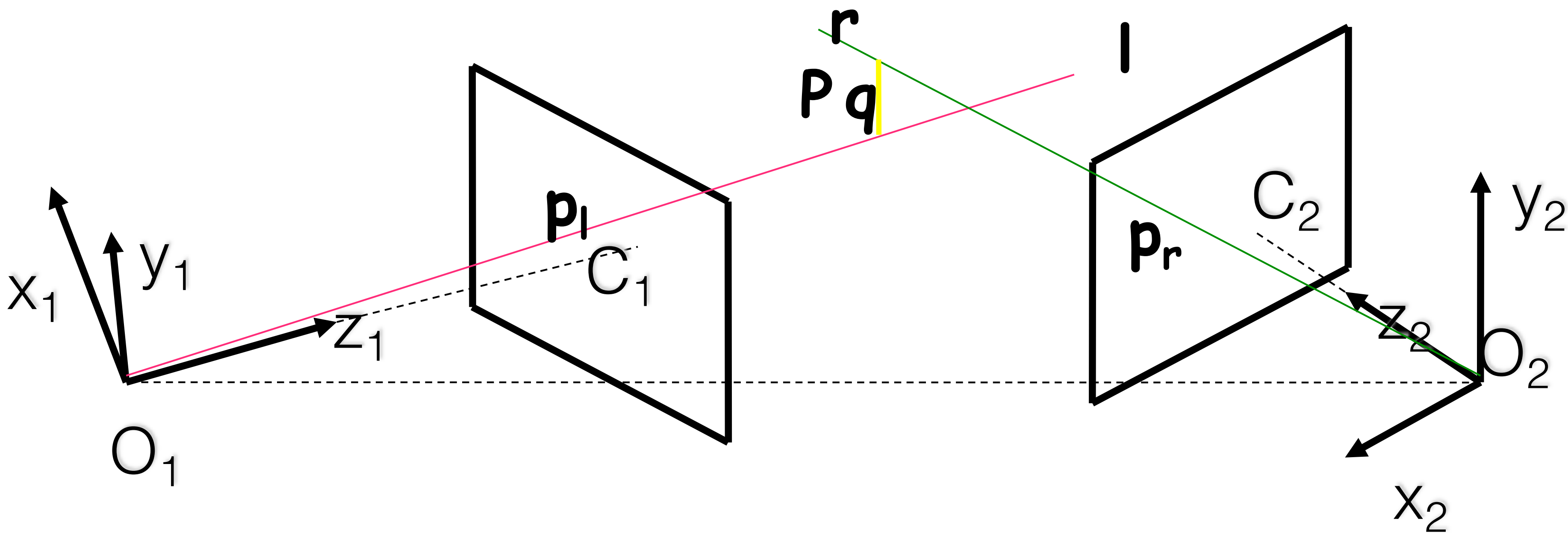
$$p = a.p_l$$

Ray r:

$$p = T + b.R^T p_r$$

Consider now a line PERPENDICULAR to both

3D Reconstruction



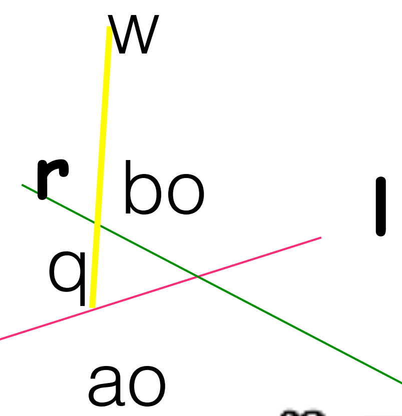
Ray l:

Ray r:

$$p = a.p_l \qquad p = T + b.R^T p_r$$

$$w \sim c.(p_l \times R^T p_r) \quad w \perp r \;\; ; w \perp l$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

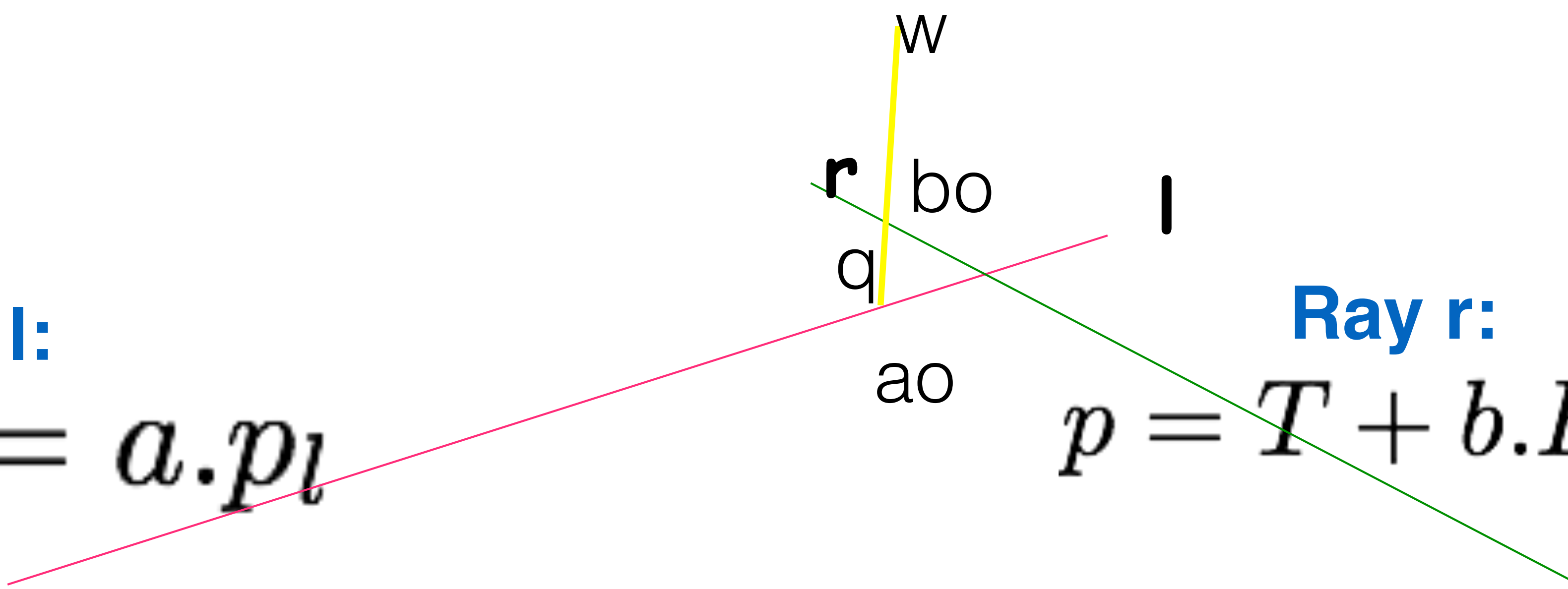
Ray r:

$$p = T + b.R^T p_r$$

$$w \sim c.(p_l \times R^T p_r) \quad w \perp r \;\; ; w \perp l$$

$$a_o p_l + \underbrace{c_o(p_l \times R^T p_r)}_{q} = T + b_o R^T p_r$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

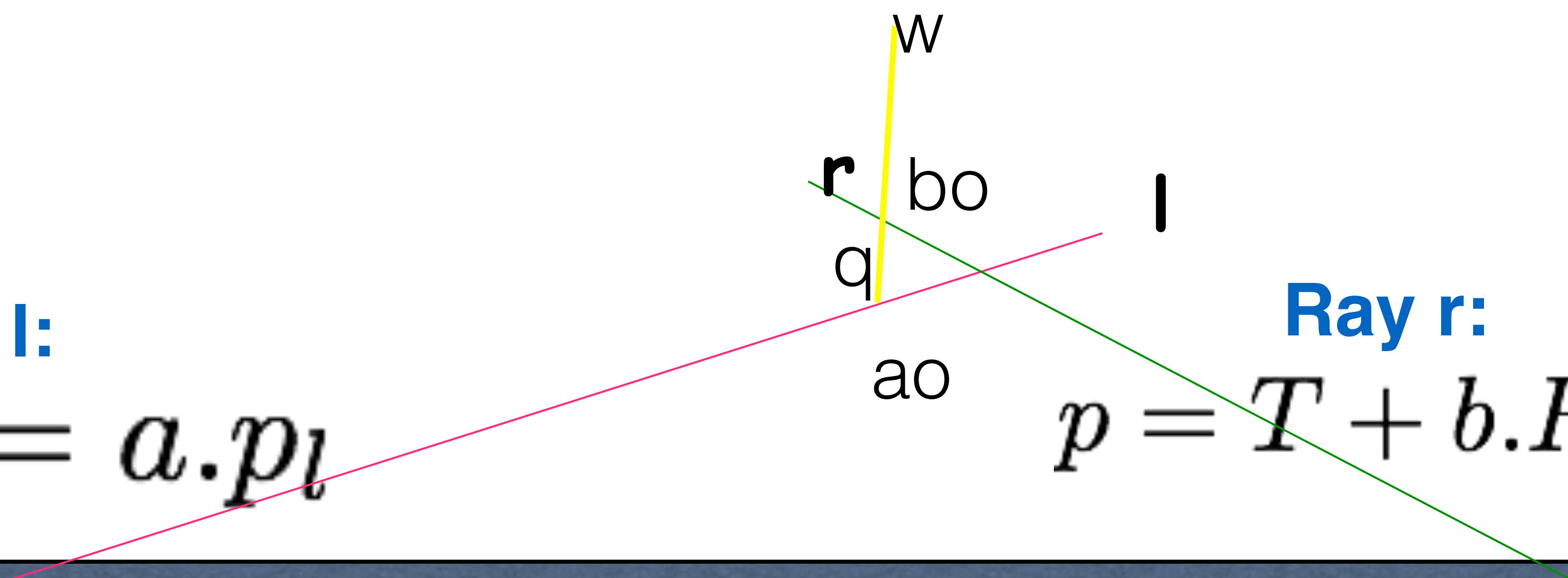
Ray r:

$$p = T + b.R^T p_r$$

$$a_o p_l + c_o(p_l \times R^T p_r) = T + b_o R^T p_r$$

$$\boxed{a_o p_l - b_o R^T p_r + c_o(p_l \times R^T p_r) = T}$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

Ray r:

$$p = T + b.R^T p_r$$

$$\boxed{a_o p_l - b_o R^T p_r + c_o (p_l \times R^T p_r) = T}$$

Solve for a_o, b_o, and c_o. How?
Choose P as the Midpoint of segment $a_o b_o$

Only <u>**INTRINSIC**</u> parameters known

3D Reconstruction

Up to a Scale Factor

3D Reconstruction

Assume:

Intrinsic parameters are known for both cameras

N >= 8 correspondences are given;

The Essential matrix was found from the above

Since we don't know the BASELINE, we cannot recover the SCALE of the scene.

Reconstruction is up to a scale factor

3D Reconstruction

4 steps:

1) Recover a normalized translation vector T

   True orientation (up to a sign), but unit length

2) Recover rotation R

3) Reconstruct Zl and Zr for each point

4) Test the signs for consistency and to recover true sign of T

Step 1: Recover T (up to a scale)

Remember the Essential Matrix:

$$E = RS$$

R is the rotation matrix

$$T \times P_l = SP_l = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \begin{bmatrix} P_{l_x} \\ P_{l_y} \\ P_{l_z} \end{bmatrix} = \begin{bmatrix} T_y P_{l_z} - T_z P_{l_y} \\ T_z P_{l_x} - T_x P_{l_z} \\ T_x P_{l_y} - T_y P_{l_x} \end{bmatrix}$$

S has rank 2 ; it depends only on T

Step 1: Recover T (up to a scale)

$$E^T E = S^T \underbrace{R^T R}_{\textbf{Identity!}} S = S^T S$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

$$S^T = \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} \qquad S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Compute the TRACE:

$$\text{Trace}(S^T S) = 2T_x^2 + 2T_y^2 + 2T_z^2 = 2||T||^2$$

$$N = \sqrt{\text{Trace}(S^T S)/2}$$

Normalization factor

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Normalize T by N: $N = \sqrt{\mathrm{Trace}(S^T S)/2}$

$$\hat{T} = \frac{T}{||T||} = \frac{T}{N}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Using the normalized translation (divide by N the matrix above):

$$\hat{E}^T \hat{E} = \begin{bmatrix} 1 - \hat{T}_x^{\,2} & -\hat{T}_x \hat{T}_y & -\hat{T}_x T_z \\ -\hat{T}_y \hat{T}_x & 1 - \hat{T}_y^{\,2} & -\hat{T}_y \hat{T}_z \\ -\hat{T}_z \hat{T}_x & -\hat{T}_z \hat{T}_y & 1 - \hat{T}_z^{\,2} \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$\hat{E}^T \hat{E} = \begin{bmatrix} 1 - \hat{T}_x{}^2 & -\hat{T}_x \hat{T}_y & -\hat{T}_x T_z \\ -\hat{T}_y \hat{T}_x & 1 - \hat{T}_y{}^2 & -\hat{T}_y \hat{T}_z \\ -\hat{T}_z \hat{T}_x & -\hat{T}_z \hat{T}_y & 1 - \hat{T}_z{}^2 \end{bmatrix}$$

Find a unit vector $\hat{T}$

This can be done from any row or column.
HOWEVER, only up to a sign since each entry is quadratic

Step 2: Find R

$$\hat{E}_i \text{ is the } i^{th} \text{ row of } \hat{E}$$

Define: $$w_i = \hat{E}_i \times \hat{T}$$

Then (Exercise!):

$$\begin{aligned} R_1 &= w_1 + w_2 \times w_3 \\ R_2 &= w_2 + w_3 \times w_1 \\ R_3 &= w_3 + w_1 \times w_2 \end{aligned}$$

Are the rows of R

Step 3: Find Zl and Zr

Given E (up to a sign factor)

- we can find T up to a sign (and scale factor)
- Once we have T we can find R

Therefore, we have 4 possibilities for (T,R), depending on the sign of E and T!

Step 3: Find Zl and Zr

Consider P expressed in the left and right coordinates: Pl and Pr

$$P_r = R(P_l - T)$$

And consider its left and right images: pl and pr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)} \qquad p_l = f_l \frac{P_l}{Z_l}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l}$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)} \xrightarrow{3} \boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l} \xrightarrow{5} \text{Pl}$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

4

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)} \xrightarrow{3} \boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l} \xrightarrow{5} P_l$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)} \xrightarrow{3} \boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

4

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l}$$

5

Pl

6

$$P_r = R(P_l - T)$$

7

Pr

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

4

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)}$$

3

$$\boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

Step 4: Check the sign of Zl and Zr

We had 4 possible pairs (R,T) and for each pair we can compute Zl and Zr

ONLY one set yields Zl and Zr positive for ALL the points:

- If BOTH < 0 for some pt
  - Change the sign of T, recompute depths
- If ONE < 0 for some pt
  - Change the sign of each entry in E, recompute R and depths
- If BOTH > 0 we have the right (R,T)

NO CALIBRATION

3D Reconstruction

Up to a PROJECTIVITY

Projective Ambiguity

Fig. 1.4. **Projective ambiguity:** *Reconstructions of a mug (shown with the true shape in the centre) under 3D projective transformations in the Z direction. Five examples of the cup with different degrees of projective distortion are shown. The shapes are quite different from the original.*

3D Reconstruction

Assume:

- No calibration is known
- N >= 8 correspondences are given;
- The Fundamental matrix was found from the above

Reconstruction is up to a projectivity

3D Reconstruction

We can recover the projection matrix for each camera up to an unknown projectivity.

Once we have these matrices, we can triangulate in projective space.

2D Projectivity

Consider the projective space P^2

Pts have THREE coordinates $(X,Y,1)'$

Recall: Given 4 correspondences between points (not three colinear) we can find the projectivity (up to a constant) between the points.

STANDARD BASIS PTS in P2

$$
\begin{aligned}
P_1 &= [1,0,0]^T \\
P_2 &= [0,1,0]^T \\
P_3 &= [0,0,1]^T \\
P_4 &= [1,1,1]^T
\end{aligned}
$$

3D Projectivity

Consider the projective space P^3

Pts have FOUR coordinates (X,Y,Z,1)'

Recall: Given 5 correspondences between points (not three colinear, not four coplanar) we can find the projectivity (up to a constant) between the points.

STANDARD BASIS PTS in P3

$$\begin{aligned}
P_1 &= [1,0,0,0]^T \\
P_2 &= [0,1,0,0]^T \\
P_3 &= [0,0,1,0]^T \\
P_4 &= [0,0,0,1]^T \\
P_5 &= [1,1,1,1]^T
\end{aligned}$$

3D Reconstruction

Choose 4 pts in one image and choose them as a basis for p2 to p2

Corresponding pts in 3D plus one more are chosen as basis for p3 to p3

Makes math easier (lots of zeros!)

Solve for M: matrix projection between p in p2 and P in P3

Impose that epipoles are in the null space of M.

Let M be the projection matrix for the left image.

Consider first 5 pts where no 3 collinear, 4 no coplanar as the standard basis sent to the standard basis plus one more point:

$$\begin{aligned} P_1 &= [1,0,0,0]^T \\ P_2 &= [0,1,0,0]^T \\ P_3 &= [0,0,1,0]^T \\ P_4 &= [0,0,0,1]^T \\ P_5 &= [1,1,1,1]^T \end{aligned}$$

$$MP_i = \rho_i p_i$$

$$\begin{aligned} p_1 &= [1,0,0]^T \\ p_2 &= [0,1,0]^T \\ p_3 &= [0,0,1]^T \\ p_4 &= [1,1,1]^T \end{aligned}$$

$$MP_1 = \rho_1 p_1$$

$$\begin{bmatrix} m11 & m12 & m13 & m14 \\ m21 & m22 & m23 & m24 \\ m31 & m32 & m33 & m34 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \rho_1 \\ 0 \\ 0 \end{bmatrix}$$

$$MP_4 = \rho_4 p_4$$

$$\begin{bmatrix} m11 & m12 & m13 & m14 \\ m21 & m22 & m23 & m24 \\ m31 & m32 & m33 & m34 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \rho_4 \\ \rho_4 \\ \rho_4 \end{bmatrix}$$

$$M = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix}$$

$$\begin{aligned} P_1 &= [1,0,0,0]^T \\ P_2 &= [0,1,0,0]^T \\ P_3 &= [0,0,1,0]^T \\ P_4 &= [0,0,0,1]^T \\ P_5 &= [1,1,1,1]^T \end{aligned}$$

$$\begin{aligned} p_1 &= [1,0,0]^T \\ p_2 &= [0,1,0]^T \\ p_3 &= [0,0,1]^T \\ p_4 &= [1,1,1]^T \\ p_5 &= [\alpha,\beta,\gamma]^T \end{aligned}$$

$$MP_5 = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \rho_5\alpha \\ \rho_5\beta \\ \rho_5\gamma \end{bmatrix}$$

$$\begin{aligned} \rho_1 &= \rho_5\alpha - \rho_4 \\ \rho_2 &= \rho_5\beta - \rho_4 \\ \rho_3 &= \rho_5\gamma - \rho_4 \end{aligned}$$

$$M = \begin{bmatrix} \rho_5\alpha - \rho_4 & 0 & 0 & \rho_4 \\ 0 & \rho_5\beta - \rho_4 & 0 & \rho_4 \\ 0 & 0 & \rho_5\gamma - \rho_4 & \rho_4 \end{bmatrix}$$

Since M is up to a scale:

$$M = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}$$

To find x, consider the projection of the center of projection:

M projects every point in P3 to P2, with the exception of the center of projection O.

M is rank 3. So O is the non trivial null space of M.

$$M.O = 0$$

$$MO = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix} \begin{bmatrix} O_x \\ O_y \\ O_z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Solving for O:

$$O_x = \frac{1}{1 - \alpha x}$$

$$O_y = \frac{1}{1 - \beta x}$$

$$O_z = \frac{1}{1 - \gamma x}$$

Repeat all the above steps for the second camera …

$$M' = \begin{bmatrix} \alpha' x' - 1 & 0 & 0 & 1 \\ 0 & \beta' x' - 1 & 0 & 1 \\ 0 & 0 & \gamma' x' - 1 & 1 \end{bmatrix}$$

$$O'_x = \frac{1}{1 - \alpha' x'}$$

$$O'_y = \frac{1}{1 - \beta' x'}$$

$$O'_z = \frac{1}{1 - \gamma' x'}$$

Now let's use the epipoles (remember that the epipoles are the images of the center of projections of the other camera) to completely find M and M'.

$$
\begin{aligned}
MO' &= \sigma e \\
M'O &= \sigma' e'
\end{aligned}
$$

$$\sigma \neq 0 \quad \sigma' \neq 0$$

$$MO' = \sigma e$$

$$M = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}$$

$$\begin{aligned} O'_x &= \frac{1}{1 - \alpha' x'} \\ O'_y &= \frac{1}{1 - \beta' x'} \\ O'_z &= \frac{1}{1 - \gamma' x'} \end{aligned}$$

$$\begin{aligned} \alpha x - 1 + 1 - \alpha' x' &= \sigma(1 - \alpha' x') e_x \\ \beta x - 1 + 1 - \beta' x' &= \sigma(1 - \beta' x') e_y \\ \gamma x - 1 + 1 - \gamma' x' &= \sigma(1 - \gamma' x') e_z \end{aligned}$$

$$
\begin{aligned}
\alpha x - \cancel{1} + \cancel{1} - \alpha' x' &= \sigma(1 - \alpha' x')e_x \\
\beta x - \cancel{1} + \cancel{1} - \beta' x' &= \sigma(1 - \beta' x')e_y \\
\gamma x - \cancel{1} + \cancel{1} - \gamma' x' &= \sigma(1 - \gamma' x')e_z
\end{aligned}
$$

$$
\begin{bmatrix} \alpha & -\alpha' & \alpha' e_x \\ \beta & -\beta' & \beta' e_y \\ \gamma & -\gamma' & \gamma' e_z \end{bmatrix} \begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix} = \begin{bmatrix} \sigma e_x \\ \sigma e_y \\ \sigma e_z \end{bmatrix}
$$

Solve for $\sigma x'$

$$\begin{bmatrix} \alpha & -\alpha' & \alpha' e_x \\ \beta & -\beta' & \beta' e_y \\ \gamma & -\gamma' & \gamma' e_z \end{bmatrix} \begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix} = \begin{bmatrix} \sigma e_x \\ \sigma e_y \\ \sigma e_z \end{bmatrix}$$

$$\begin{bmatrix} p_5 & p_5' & v \end{bmatrix} \begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix} = \sigma e$$

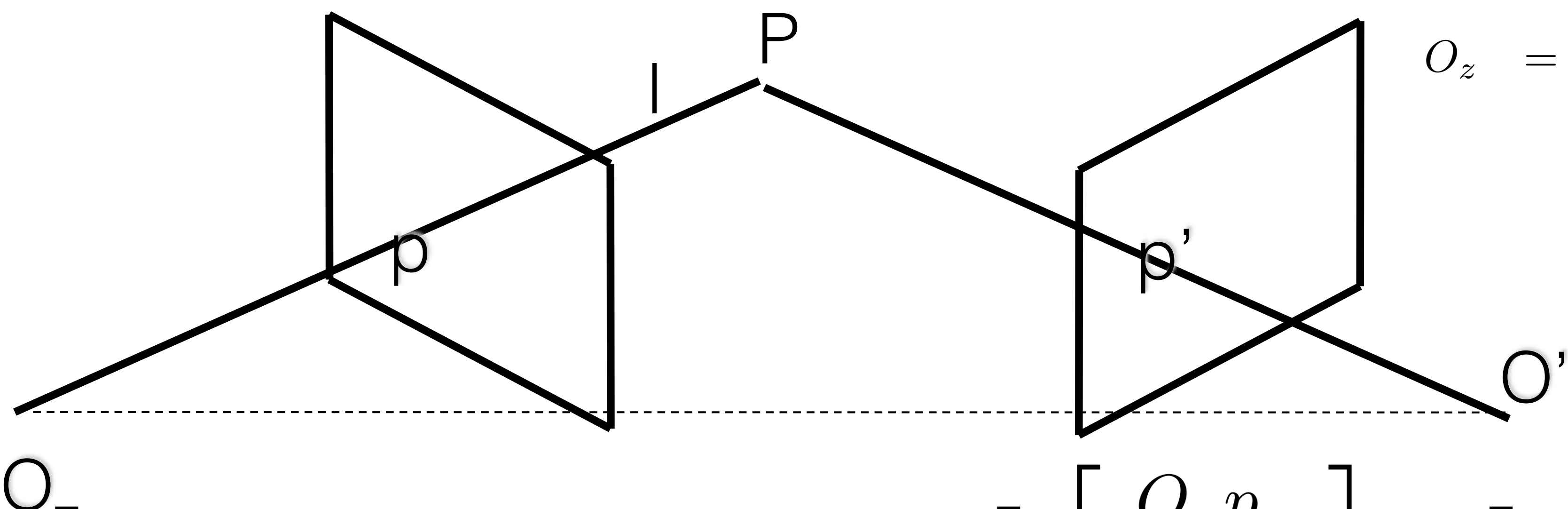
$$\cancel{\sigma} x' = \cancel{\sigma} \frac{e^T (p_5 \times p_5')}{v^T (p_5 \times p_5')}$$

And similarly for x:

$$x = \frac{e'^T(p_5 \times p'_5)}{v'^T(p_5 \times p'_5)}$$

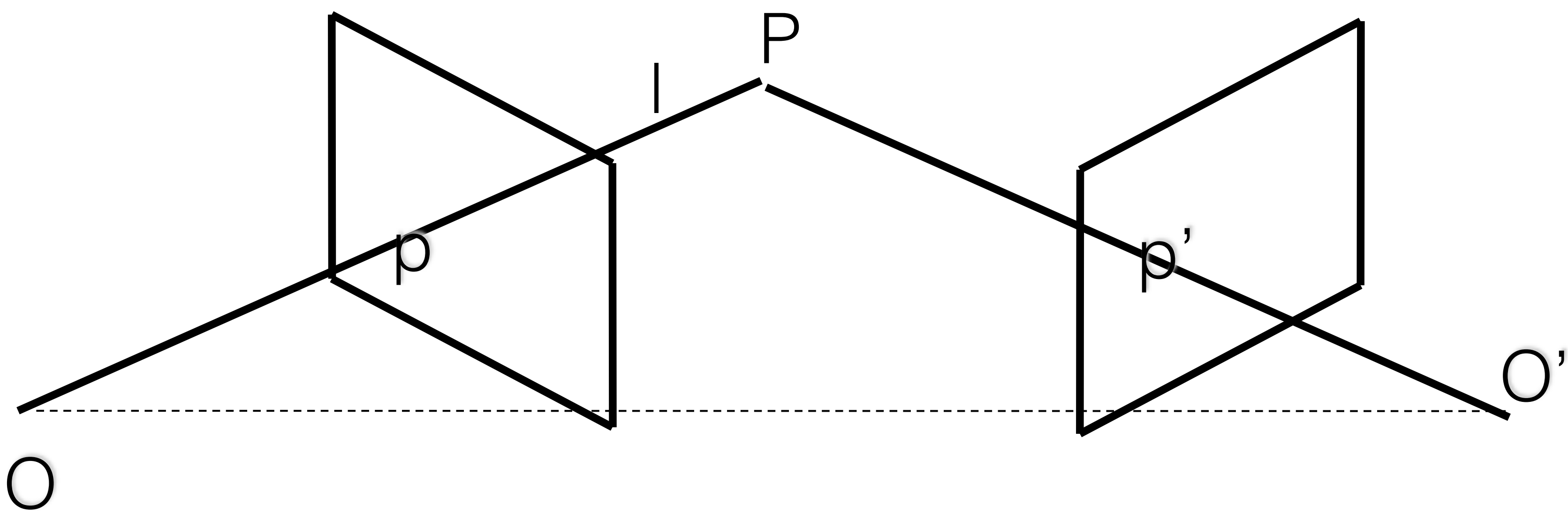
Now we have M and M', O and O', so we can reconstruct any point in P3 given its corresponding images p and p'.

$$O_x = \frac{1}{1-\alpha x}$$
$$O_y = \frac{1}{1-\beta x}$$
$$O_z = \frac{1}{1-\gamma x}$$



$$\begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix} \begin{bmatrix} O_x p_x \\ O_y p_y \\ O_z p_z \\ 0 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

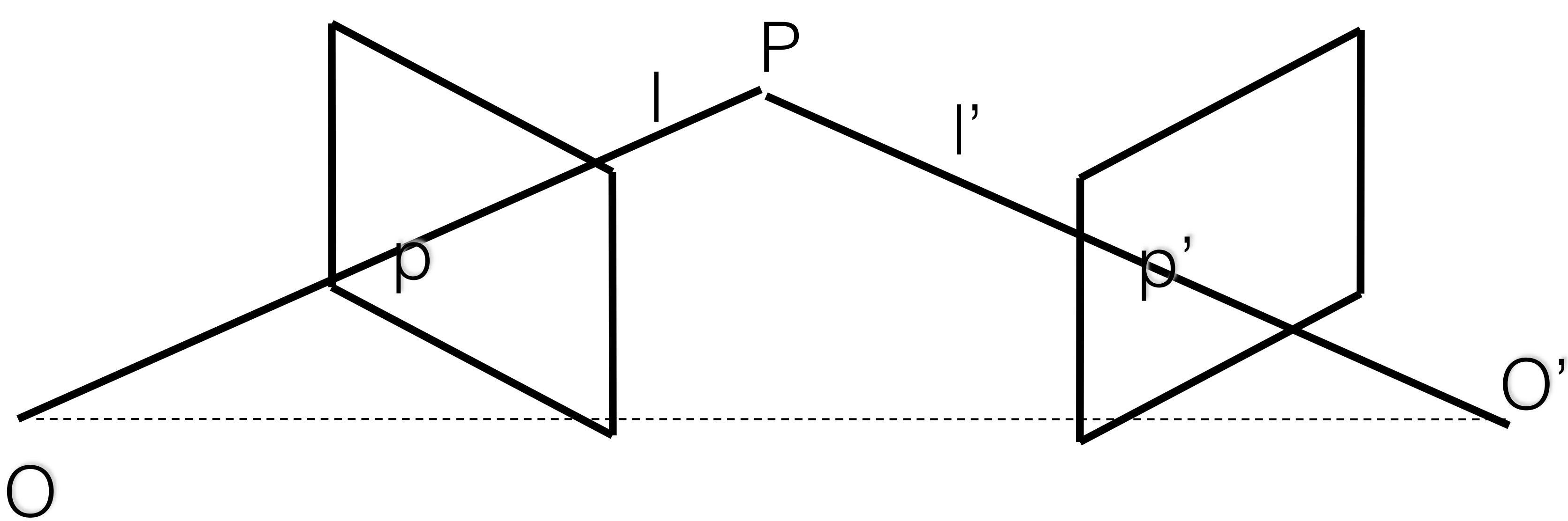
Now we have M and M', O and O', so we can reconstruct any point in P3 given its corresponding images p and p'.



$$\lambda O + \mu [O_x p_x, O_y p_y, O_z p_z, 0]^T$$

Line l that passes through O and p

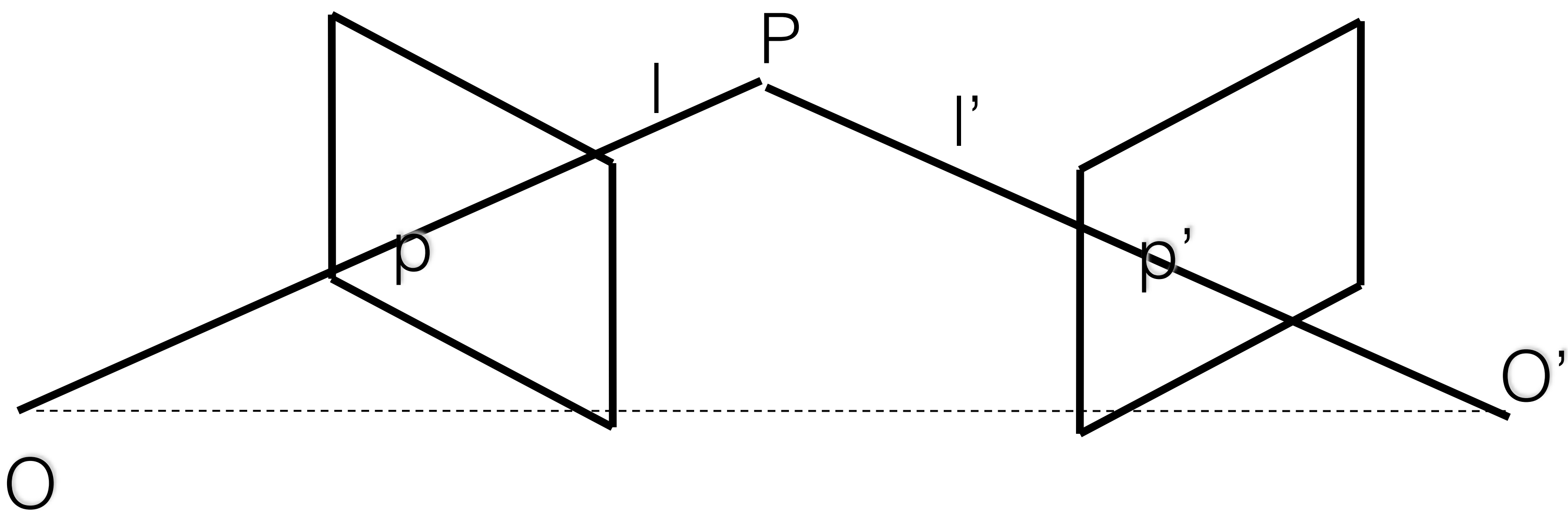
Similarly, for p’



$$\lambda' O' + \mu' [O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

Line l’ that passes through O’ and p’

P lies in the intersection of l and l'



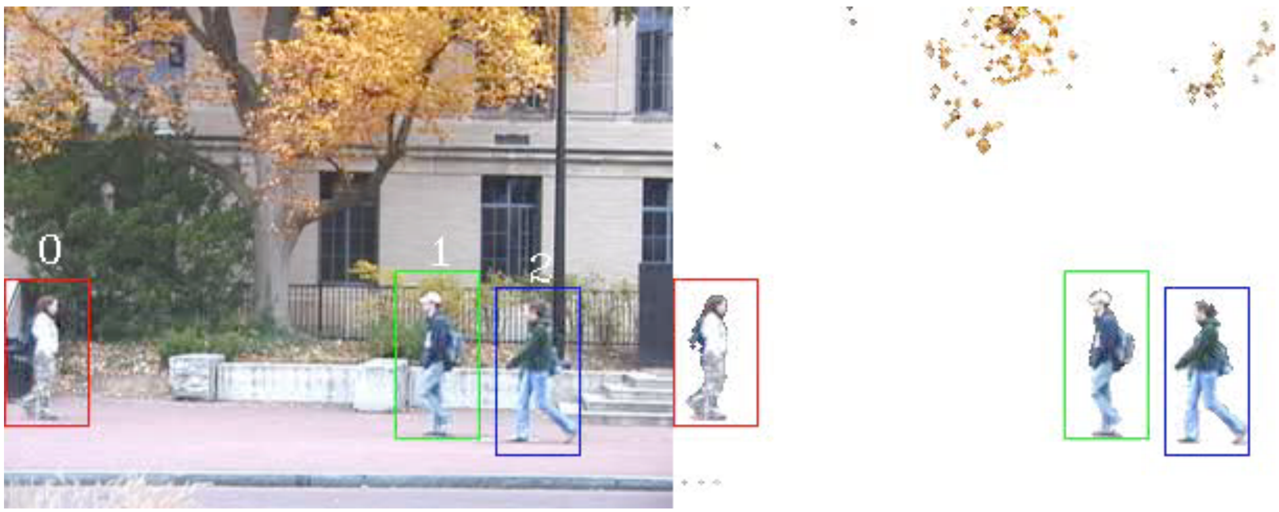
$$\lambda O + \mu[O_x p_x, O_y p_y, O_z p_z, 0]^T = \lambda' O' + \mu'[O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

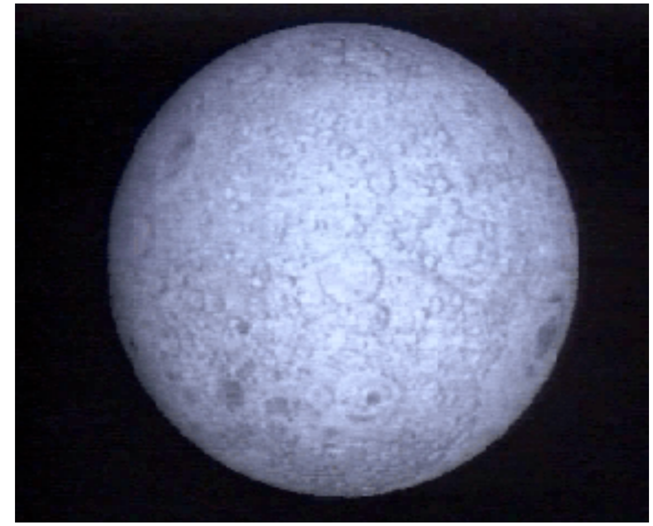
$$\lambda O + \mu[O_x p_x, O_y p_y, O_z p_z, 0]^T = \lambda' O' + \mu'[O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

$$\begin{bmatrix} O_x & O_x p_x & -O'_x & -O'_x p'_x \\ O_y & O_x p_y & -O'_y & -O'_y p'_y \\ O_z & O_z p_z & -O'_z & -O'_z p'_z \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \lambda' \\ \mu' \end{bmatrix} = 0$$

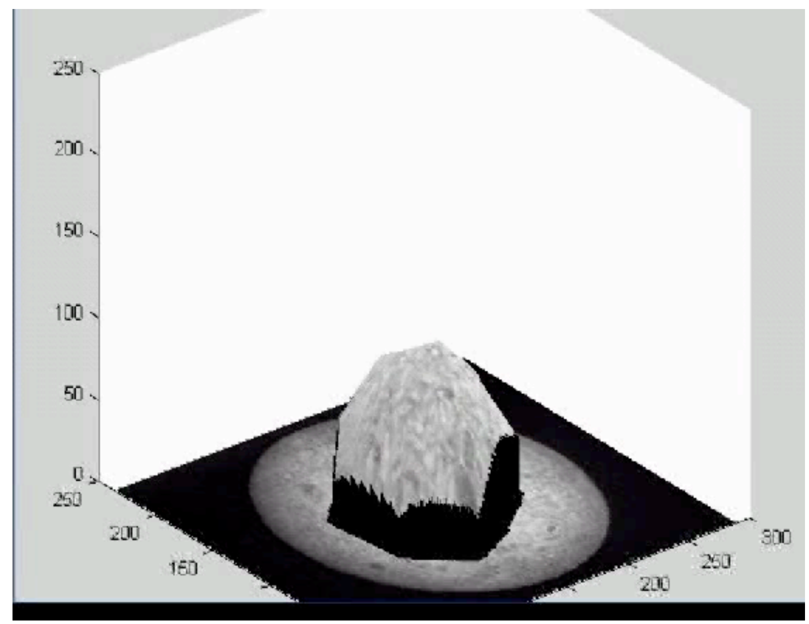
Compute its SVD: UDV'.
The solution is given by the column of V corresponding to the smallest ev.

Motion


0
1
2




1
2
3
4


250
200
150
100
50
0
250
200
150
300
250
200

Process of images over time.

Image sequence:

An image sequence is a series of N images, or frames, acquired at discrete time instants $t_k = t_o + k\Delta t$, where Δt is a fixed time interval and $k=0,1,\ldots,N-1$

Assuming that illumination does not change:

Image changes are due to the RELATIVE MOTION between the scene and the camera.

There are 3 possibilities:

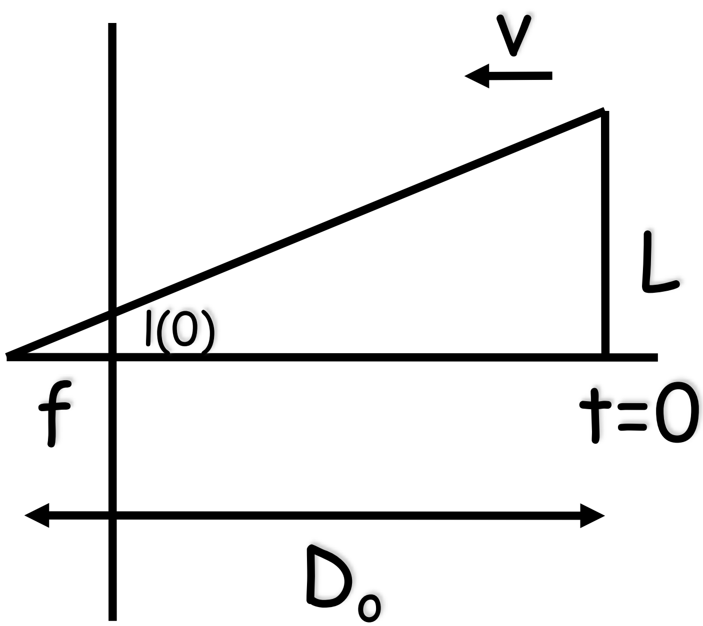
- Camera still, moving scene
- Moving camera, still scene
- Moving camera, moving scene

Visual Motion

Allows us to compute useful properties of the 3D world, with very little knowledge.

Example: Time to collision

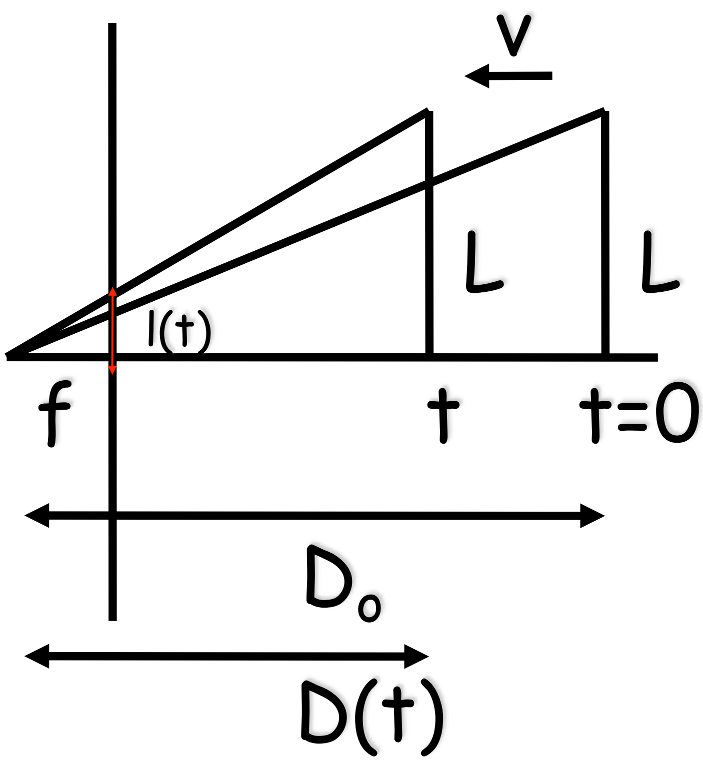
Time to Collision



An object of height L moves with constant velocity v:

- At time t=0 the object is at:
  - $D(0) = D_o$

Time to Collision



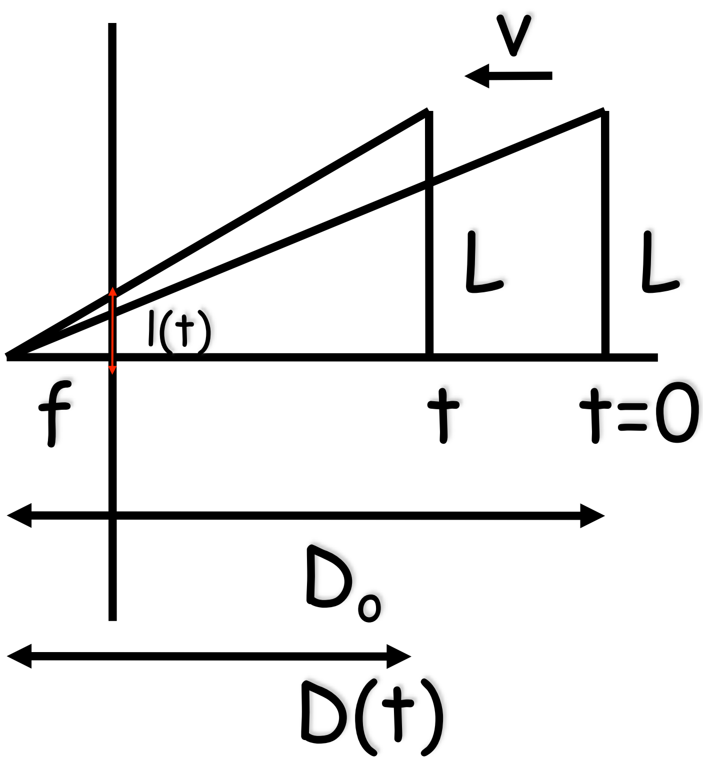
An object of height L moves with constant velocity v:

- At time t=0 the object is at:
  - $D(0) = D_o$
- At time t it is at
  - $D(t) = D_o - vt$
- It will crash with the camera at time:
  - $D(\tau) = D_o - v\tau = 0$
  - $\tau = D_o/v$

Time to Collision

The image of the object has size l(t):

$$l(t) = \frac{fL}{D(t)}$$

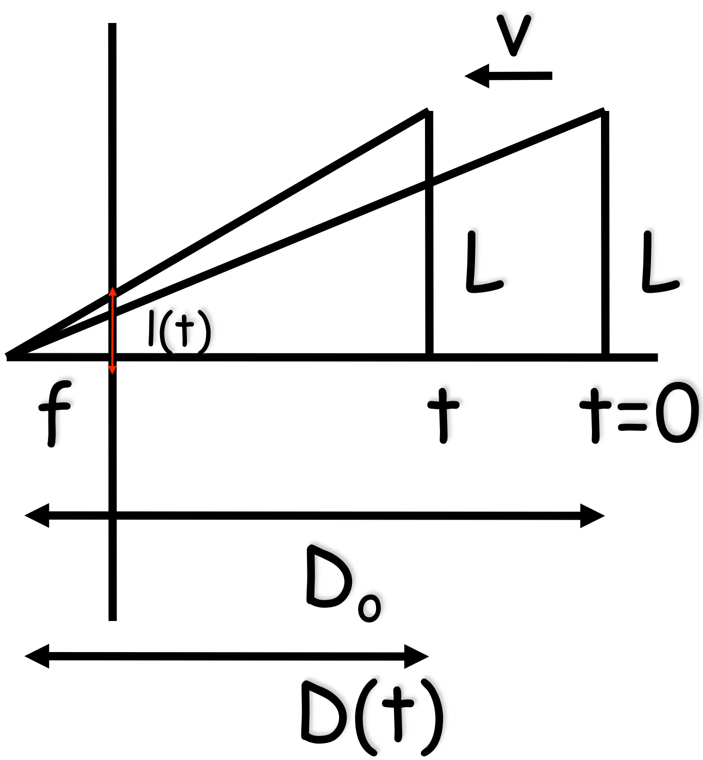


Taking derivative wrt time:

$$l'(t) = \frac{dl(t)}{dt} = fL\frac{d(1/D(t))}{dt}$$

$$l'(t) = fL\frac{-1}{D^2(t)}\frac{d(D(t))}{dt}$$

Time to Collision



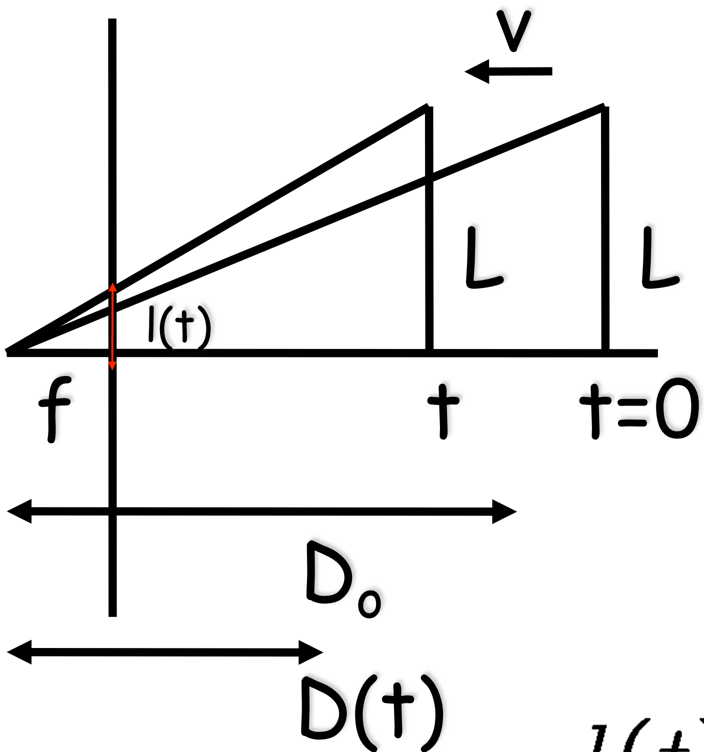
$$l'(t) = fL\frac{-1}{D^2(t)}\frac{d(D(t))}{dt}$$

$$D(t) = D_o - vt$$

$$\frac{d(D(t))}{dt} = -v$$

$$l'(t) = fL\frac{v}{D^2(t)}$$

Time to Collision



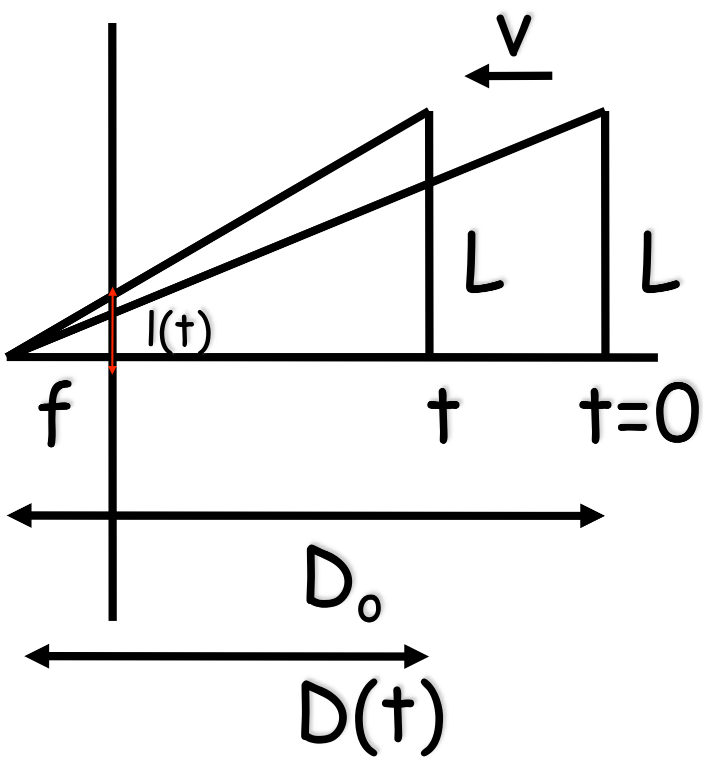
$$l'(t) = fL\frac{v}{D^2(t)}$$

$$l(t) = \frac{fL}{D(t)}$$

And their ratio is:

$$\frac{l(t)}{l'(t)} = \frac{fL}{D(t)}\frac{D^2(t)}{fLv} = \frac{D(t)}{v} = \tau$$

Time to Collision



$$l'(t) = fL\frac{v}{D^2(t)}$$

$$l(t) = \frac{fL}{D(t)}$$

Can be directly measured from image

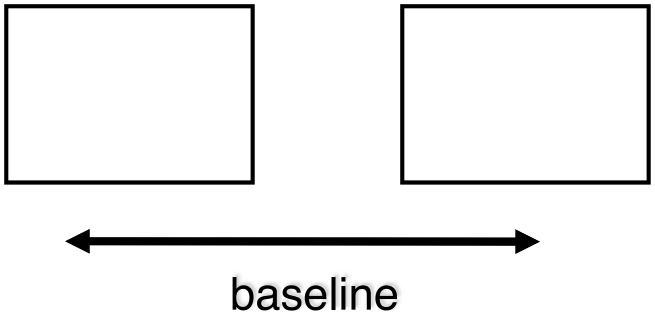
And time to collision:

$$\tau = \frac{l(t)}{l'(t)}$$

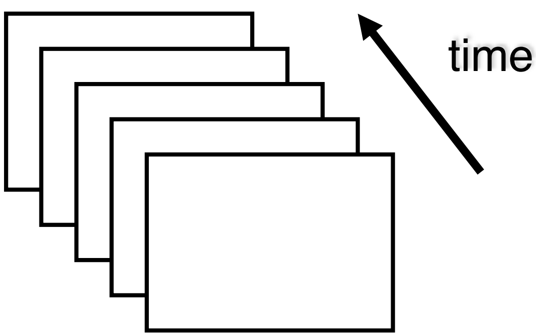
Can be found, without knowing L or D_o or v !!

Comparison between Motion Analysis and Stereo

Stereo: Two or more frames



Motion: N frames



- Baseline is usually larger in stereo than in motion:
  - Motion disparities tend to be smaller
- Stereo images are taken at the same time:
  - Motion disparities can be due to scene motion
  - There can be more than 1 transf. btw frames

Motion Analysis Problems

Correspondence Problem

Track corresponding elements across frames

Reconstruction Problem

Given a number of corresponding elements, and camera parameters, what can we say about the 3D motion and structure of the observed scene?

Segmentation Problem

What are the regions of the image plane which correspond to *different* moving objects?

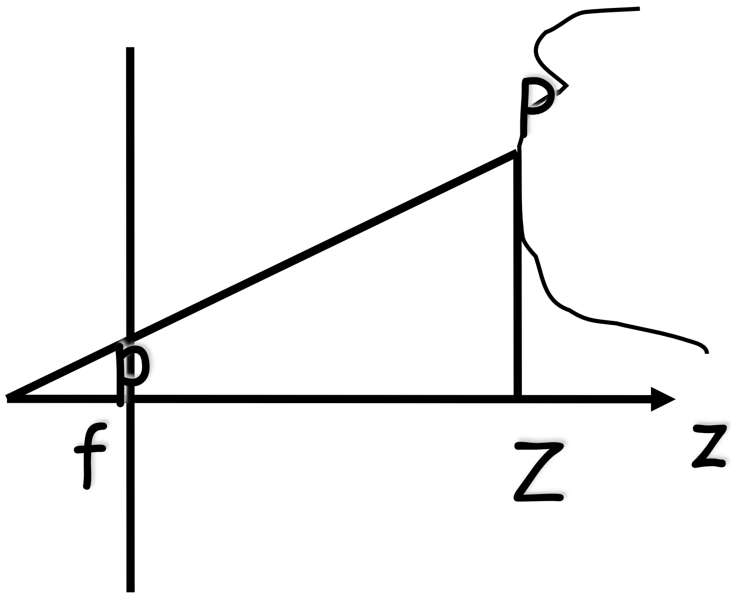
Motion Field (MF)

The MF assigns a velocity vector to each pixel in the image.

These velocities are INDUCED by the RELATIVE MOTION btw the camera and the 3D scene

The MF can be thought as the *projection* of the 3D velocities on the image plane.

Consider a 3D point P and its image:



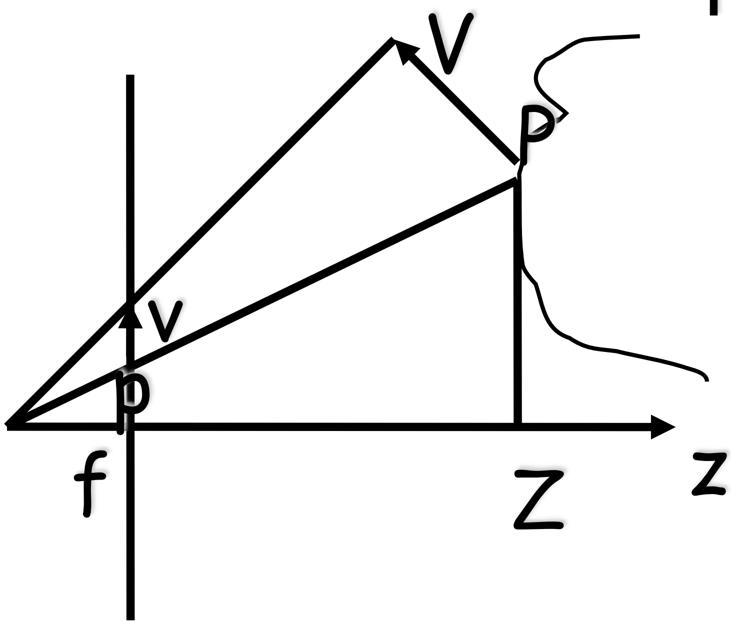
$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$p = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$

Using the pinhole camera equation:

$$p = \frac{fP}{Z}$$

Let things move:



The relative velocity of P wrt camera:

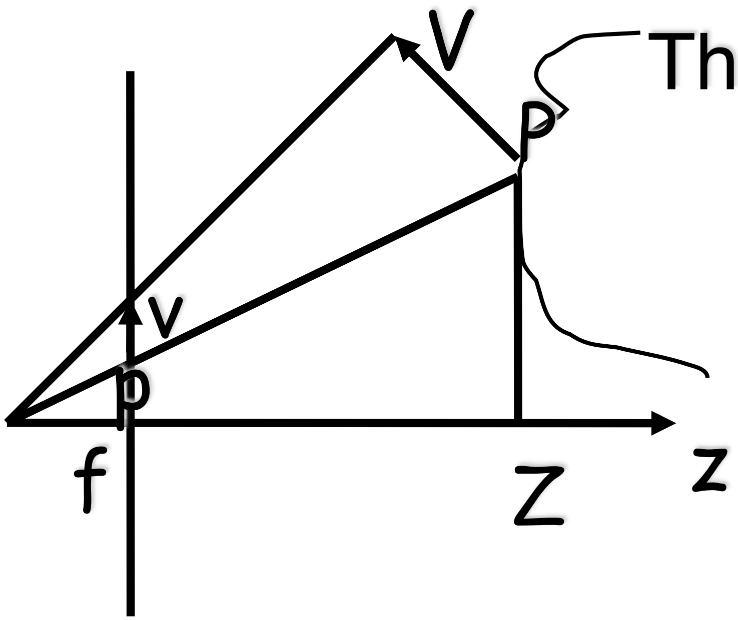
$$V = -T - \omega \times P$$

Translation velocity

Rotation angular velocity

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \qquad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

3D Relative Velocity:



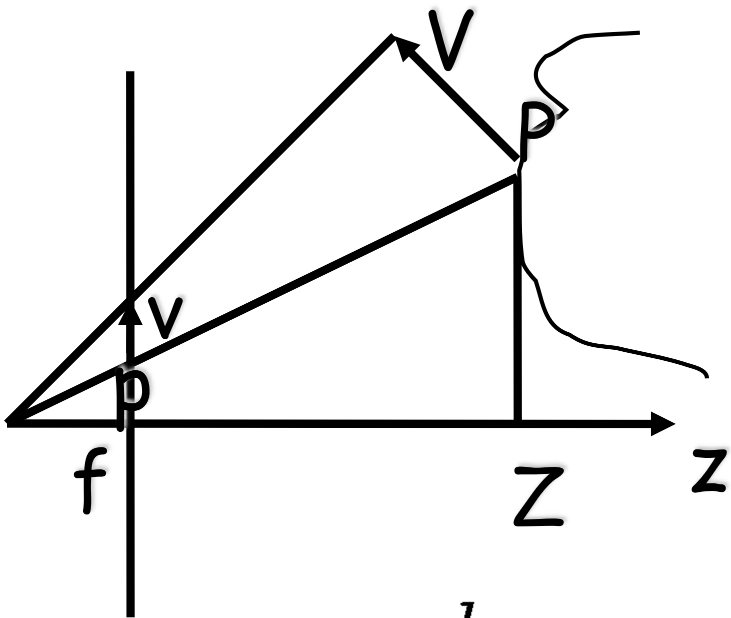
The relative velocity of P wrt camera:

$$V = -T - \omega \times P$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{aligned} V_x &= -T_x - \omega_y Z + \omega_z Y \\ V_y &= -T_y - \omega_z X + \omega_x Z \\ V_z &= -T_z - \omega_x Y + \omega_y X \end{aligned}$$

Motion Field: the velocity of p



$$p = \frac{fP}{Z}$$

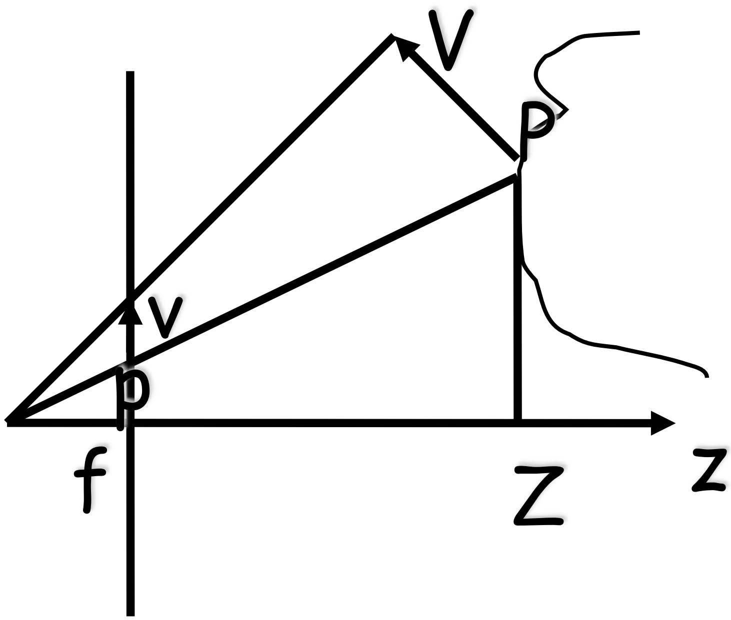
Taking derivative wrt time:

$$\frac{dp}{dt} = v = \frac{d\frac{fP}{Z}}{dt}$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[\frac{dP}{dt}.Z - P.\frac{dZ}{dt}\right]$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}[V.Z - P.V_z]$$

Motion Field: the velocity of p

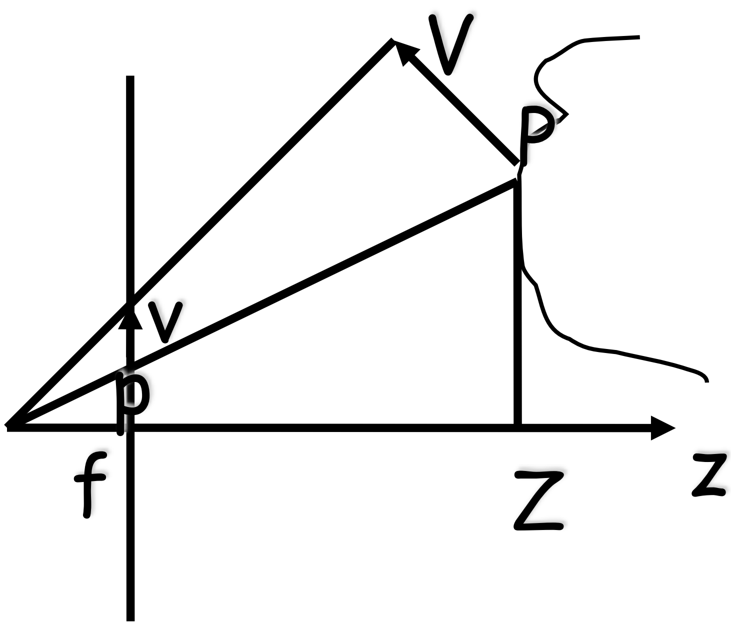


$$\frac{dp}{dt} = v = \frac{f}{Z^2}[V.Z - P.V_z]$$

$$p = \frac{fP}{Z} \qquad P = \frac{pZ}{f}$$

$$\boxed{v = f\frac{V}{Z} - p\frac{V_z}{Z}}$$

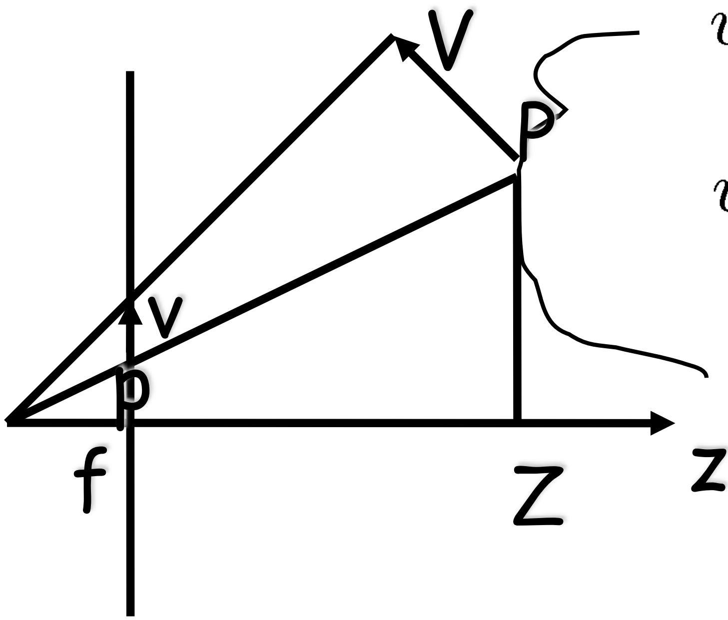
Motion Field: the velocity of p



$$\boxed{v = f\frac{V}{Z} - p\frac{V_z}{Z}}$$

$$\begin{aligned} v_x &= f\frac{V_x}{Z} - x\frac{V_z}{Z} \\ v_y &= f\frac{V_y}{Z} - y\frac{V_z}{Z} \\ v_z &= f\frac{V_z}{Z} - f\frac{V_z}{Z} = 0 \end{aligned}$$

Motion Field: the velocity of p



$$
\begin{aligned}
v_x &= f\frac{V_x}{Z} - x\frac{V_z}{Z} \\
v_y &= f\frac{V_y}{Z} - y\frac{V_z}{Z}
\end{aligned}
$$

$$
\begin{aligned}
V_x &= -T_x - \omega_y Z + \omega_z Y \\
V_y &= -T_y - \omega_z X + \omega_x Z \\
V_z &= -T_z - \omega_x Y + \omega_y X
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
v_x &= \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x x y}{f} - \frac{\omega_y x^2}{f} \\
v_y &= \frac{T_z y - T_y f}{Z} + \omega_x f - \omega_z x - \frac{\omega_y x y}{f} + \frac{\omega_x y^2}{f}
\end{aligned}
}
$$